



saschavanessen@gmail.com

Python Course

Week 3:
Loops and strings



Workshop Overview

1. Writing your first program
2. Making choices and reuse code
3. Loops and strings
4. Files and lists
5. Dictionaries and tuples

Acknowledgments: Structure of the workshop follows the book “Python for informatics” by Charles Severance. Several examples are from this book or the accompanying slides.



Assignments last week

- Lootjes 1
- Lootjes 2
- Bread 2

Loops





While loop

```
n = 10  
while n >= 0:  
    print(n)  
    n = n - 1  
print("Blast off!")
```

initialization

condition

update



Declaration of a variable

```
>>> x = x + 1
```

← declaring x

You can't update/use a variable that doesn't exist yet!

```
>>> x = 0
```

```
>>> x = x + 1    or    >>> x += 1
```

← updating x



What happens in a loop?

- 1) Evaluate the condition:
True or False?
- 2) If the condition is False:
exit the loop and continue
with the rest of the code
- 3) If the condition is True:
execute the body of the
loop and go back to step 1





99 Bottles of Beer

99 bottles of beer on the wall, 99 bottles of beer.
Take one down, pass it around, 98 bottles of beer on the wall...
98 bottles of beer on the wall, 98 bottles of beer.
Take one down, pass it around, 97 bottles of beer on the wall...

...

No more bottles of beer on the wall, no more bottles of beer.
Go to the store and buy some more, 99 bottles of beer on the wall...

Try this script and fix the error: `bottles-of-beer-contains-error.py`
But don't run it yet!



Infinite loops

- Try: `bottles-of-beer-infinite-loop.py`
- With CTRL+C you can stop the program



- Try your solution of `bottles-of-beer-contains-error.py`



Infinite loops

```
n = 99
while n >= 0:
    if n > 0:
        print(n, "...", n, "...")
        n = n - 1
        print("...", n, "...")
    else:
        print("No more...")
        print("...")
```

initialization

condition

no update!!

You need to make sure the loop can end!



Useful infinite loops examples

- Let a sensor measure the temperature endlessly
- Ask a user for input until an answer is correct:

```
while True:
    line = input("Say 'when' ... ")
    if line == 'when':
        break
    print("Not yet I guess!")
print("Done!")
```



Continue and break

```
while True:
    inp = input("Enter temperature: ")
    try:
        temp = float(inp)
        break
    except:
        print("That was not a number!")
        continue
```



For loops

Initialization, condition
and update in one!

applepie.py

```
groceries = ["apples", "flour", "butter"]  
for item in groceries:  
    print("Don't forget the", item)
```

snoepjes.py

```
for i in range(10):  
    print("Ik heb", i, "snoepjes!")
```



Count items

```
total = 0
for item in [32, 45, 8, 10, 25]:
    total = total + item
print("Total:", total)
```



Maximum

```
largest = None
print('Largest before:', largest)

for num in [3, 41, 12, 9, 74, 15]:
    if largest is None or num > largest:
        largest = num
    print('Loop:', num, largest)

print('Largest after:', largest)
```



Assignment! Count positives

- Write a program that counts all the positive numbers in the list

[10, 4, -50, 9, -88, -12, 36]

- Hints:
 - for loop
 - a variable to count
 - if statement to check if the number is positive



Wrap-up

A loop usually has the following:

- A variable is declared and initialized (counter or something finite)
- Evaluation (the condition) of variable to end the loop
- Something is done in the loop (body)

Strings



Strings

- A string is a sequence



```
>>> fruit = "banana"
>>> letter = fruit[1]
>>> first_letter = fruit[0]
```

b	a	n	a	n	a
0	1	2	3	4	5



Length and indices

```
>>> fruit = "banana"
```

- Length:

```
>>> length = len(fruit)
```

- Get the last index:

```
>>> fruit[length]
```

```
>>> fruit[length-1]
```

- Python way:

```
>>> fruit[-1]
```



Letter by letter: while loop

```
fruit = "apple"  
index = 0  
while index < len(fruit):  
    letter = fruit[index]  
    print(letter)  
    index = index + 1
```

(index = index + 1 → index += 1)



Letter by letter: for loop

```
fruit = "orange"  
for letter in fruit:  
    print(letter)
```

That's it! :)

Slices

```
>>> s = "orange"
>>> s[0:3]    # ora
>>> s[:3]     # ora
>>> s[3:]     # nge
>>> s[3:3]    # ''
>>> s[2:5]    # ang
>>> s[:]      # orange
```





Mutability

```
>>> x = "COOKIES"
```

This will give an error:

```
>>> x[0] = "B"
```

Solution, use slices:

```
>>> new = x[:3] + L
```

```
>>> print(new)
```




Assignment! Count letters

- Write a program that counts all the a's in "banana"
- Hints:
 - for loop
 - a variable to count
 - if statement to check if the letter is an a
 - Take a look at the previous assignment (count positives)



“in” and string comparison

- Check if a letter or substring is in a string

```
>>> 'a' in 'banana'
```

```
>>> 'ana' in 'banana'
```

- Comparisons

```
>>> 'mouse' == 'mouse'
```

```
>>> 'Mouse' == 'mouse'
```

```
>>> 'elephant' > 'cow'
```

```
>>> 'dinosaur' < 'cat'
```

(like the order of a dictionary)



String methods

- `count()`
- `upper()` / `lower()`
- `find()`
- `startswith()`
- `strip()`

```
>>> animal = 'cowww'
```

```
>>> animal.count('w')
```

```
>>> animal.upper()
```

```
>>> help(str.upper)
```

Parsing strings

```
>>> line = 'From: Sascha <saschavanessen@gmail.com>'
>>> line.find("@")
28
```

```
>>> start = line.find("<")
>>> end = line.find(">")
>>> line[start:end]
'<saschavanessen@gmail.com'
```

```
>>> line[start+1:end]
saschavanessen@gmail.com
```





More on strings

- Chapter 6
 - Format operator (not in these slides)
- Documentation string methods
 - <https://docs.python.org/3/library/string.html>



Assignment! Guessing Game

- Write a program that generates a random number between 1 and 10
- Let the user guess which number the computer had in mind
- The user can guess at most 5 times
- Give hints “higher” and “lower” when the guessed number is too high or too low
- Partial code is available on the wiki ([guess-game.py](#))



To be continued!

- More practice:
 - Exercises in chapter 5 and 6 of the book
- Next week: Files and lists
- See you next week!



Icons Acknowledgements

- <https://www.flaticon.com/authors/smashicons>
- <http://www.freepik.com>



Shortcuts

- Terminal

- Up previous commands
- Tab autocomplete
- cd .. one folder up

- Editor

- Ctrl-/ or Cmd-/ comment on/off
- Tab indent forward
- Shift-Tab indent backwards